DOI:
10.1016/j.robot.2016.02.002

Document Version
Peer reviewed version

The Study of Model Predictive Control Algorithm Based on the Force/Position Control Scheme of the 5-DOF Redundant Actuation Parallel Robot

Shuhuan Wen[1], Guiqian Qin[1], Baowei Zhang[1], H.K. Lam[2], Yongsheng Zhao[3] and Hongbin Wang[1]

[1]Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao, China

[2]Department of Informatics, Kings College London, Strand, London, WC2R 2LS, United Kingdom

[3]Parallel Robot and Mechatronic System Laboratory of Hebei Province and Key Laboratory of Advanced Forging&Stamping Technology and Science of Ministry of National Education, Yanshan University, Qinhuangdao, 066004, China

Email: swen@ysu.edu.cn, yanshanqgq@163.com, 458110801@qq.com, hak-keung.lam@kcl.ac.uk, yszhao@ysu.edu.cn, hb_wang@ysu.edu.cn

Abstract

Redundant actuated parallel robot is a multi-input and multi-output (MIMO) system which usually works in an uncertain environment. In this paper, the force/position hybrid control structure of 6PUS-UPU redundant actuation parallel robot is designed, proportional-integral (PI) and model predictive control (MPC) cascade control strategy are used in the redundant branch of 6PUS-UPU redundant actuation parallel robot. The MPC algorithm is used in the current loop of the permanent magnet synchronous motor (PMSM) to restrain the motor parameter uncertainty and external disturbances influence on motor control. The MATLAB/ADAMS joint simulation method based on virtual 6PUS-UPU redundant actuation parallel robot prototype is used to test the performance of the proposed control strategy. The performance of proposed PI-MPC control strategy is compared with the traditional PI-PI control strategy. The simulation results show that the MPC controller can improve the tracking ability of the motor torque, guarantee the system robustness under the parameter variations and load disturbance environment.

Index Terms

Force/position hybrid control, parallel robot, redundant actuation, PMSM, MPC controller.

I. Introduction

Parallel robot has high rigidity, high bearing capacity, high accuracy and good dynamic performance, which makes itself gradually develop into a manufacturing star in the field of numerical control processing. However, the parallel robot is an MIMO system and works in an uncertain environment, there are many problems to be solved, such as the strong coupling of parallel robot between each branch, small work space, a large number of singular configuration, etc. These problems will result in large structure internal force, and the machine will be damaged. Redundant actuation as an effective solution is introduced. Redundancy in the parallel robot is divided into the movement redundancy and actuation redundancy [1]–[3], Movement redundancy means to add extra joints to increase the freedom of movement [4], while actuation redundant just increases the number of joints and cannot influence the degree of freedom of the machine. Comparing with the ordinary parallel robot, the parallel robot with redundant actuation can optimize the load distribution between the actuators, reduce the energy consumption of each independent actuator, avoid driving force mutation, and improve force transmission properties of homogeneity and the rigidity of the agency (active stiffness). Thus the parallel robot with redundant actuation can obtain a better dynamic performance, higher stiffness and greater carrying capacity [5], [6]. In recent years, the redundant actuation parallel robot has widely been studied by many researchers. The authors of the work in [1]–[3] proposed the main functions of the redundant actuation to optimize some performance indexes, such as power, energy consumption and the internal forces. The authors of the work in [7]–[9] developed a method of the redundant actuation which can improve the dynamic performance. The author of the work in [10] proposed a novel 6PUS-UPU redundant actuation parallel robot. The author of the work in [11] proposed two methods to improve the position and force interaction control accuracy. The first approach is a position-based sliding mode impedance control which converts the target impedance into a desired position trajectory to be tracked, and the second one is established on the basis of a proportional-integral type of sliding function of the impedance measure error. The author of the work in [12] proposed a method which both contact force exerted by the manipulator and the position of the end-effector in contact with the surface are controlled.

The current loop control method of PMSM mainly uses PI regulator, hysteretic control, sliding mode variable structure control, etc. PI controller is a linear controller and the proportional gain can improve the dynamic performance of system. However, excessive gain will result in noise, overshoot and oscillation, which has a negative influence on the stability of system. It is very difficult to trade off between the rapid response and stability in practical applications. Hysteretic control offers quick responses, but this kind of control mode usually leads to large ripple of response. The switching frequency is not fixed and it is not suitable for high performance control cases. Sliding mode variable structure control method demonstrates a ”chattering” problem [13]–[15]. The authors of the work in [16] introduced the differential feedback control into the PMSM servo system. The system offers a quick response speed and obviously suppresses well the overshoot. The authors of the work in [17] also did a similar research, but to some extent, the introduction of differential item could affect the system stability. The authors of the work in [18] used the feedforward method to compensate the cross coupling of the stator voltage, which fully realized the decoupling of the direct axis and quadrature axis(dq axis) in order to improve the current dynamic performance. The authors of the work in [19] developed the method of the voltage feedforward decoupling control of the current loop. The control performance of this method is influenced by the accuracy of the motor parameters and has poor robustness. The authors of the work in [20] developed a new predictive current control (PCC) for the PMSM drivers. The proposed control method obtains the same rapid torque dynamics as direct torque control (DTC) schemes, but it has smaller magnitude of ripples. The authors of the work in [21] developed a robust probational-integral-derivative (PID) control scheme for the PMSM by using a genetic searching approach.

In actual application, because of the influence of the temperature and other uncertain factors, some parameters of the system model can be changed and result in the model mismatch, which bring huge influence on the control performance of the whole system. The above control methods could not deal with it. So we propose a model predictive control algorithm in this paper.

MPC was proposed in the 1970s, which has been the most widely used multivariable control algorithm in the chemical process industries and other research areas. MPC is suitable for many kinds of problems, and

demonstrates the advantages especially when dealing with the problems subject to [22]: 1) a large number of manipulated and controlled variables, 2) constraints imposed on both the manipulated and controlled variables, 3) change control objectives and/or equipment (sensor/actuator) failure, 4) time delays. MPC uses the online rolling optimization and feedback correction strategy. The online rolling optimization strategy can compensate the uncertainty influence caused by model mismatch and interference so as to improve the control effect of the system [23]–[25]. After a series of future control variables are determined by the optimization, MPC uses the detected error between actual output and the model prediction output to realize feedback correction to remedy the defect. So the states of the controlled plant is prevented from deviating from the ideal states because of the model mismatch or environment interference. At the next sampling time, the detected actual output modifies the MPC output. And then a new optimization is performed. This method can overcome the uncertainty of the system, improve the robustness of the system [26]–[28]. The authors of the work in [29] compared the Forced Machine Current Control (FMCC) with the Model Predictive Direct Torque Control (MPDTC) and Model Predictive Direct Current Control (MPDCC) for MV induction motor drives. The authors of the work in [30] developed the Model Predictive Direct Speed Control (MP-DSC), which overcomes limitations of cascaded linear controllers. The authors of the work in [31], [32] developed an MPC scheme to control the speed of a PMSM drive system. The performance of the proposed controller is compared with a classical PI controller and tested by MATLAB/ADAMS simulation platform. The results show that more accurate tracking performance of the PMSM has been obtained.

For PMSM current loop control problem, we introduce the MPC algorithm to the motor control of 6PUS-UPU redundant actuation parallel robot in this paper. This algorithm can effectively solve the problems caused by the motor parameters uncertainty and external disturbance, achieve quick tracking for current orders without overshoot.

In this paper, PMSM space voltage vector control simulation platform is built, and MPC strategy is used in the redundant branch of 6PUS-UPU redundant actuation parallel robot. The performance of MPC controller and PI controller are compared, and the results of simulation demonstrate that MPC

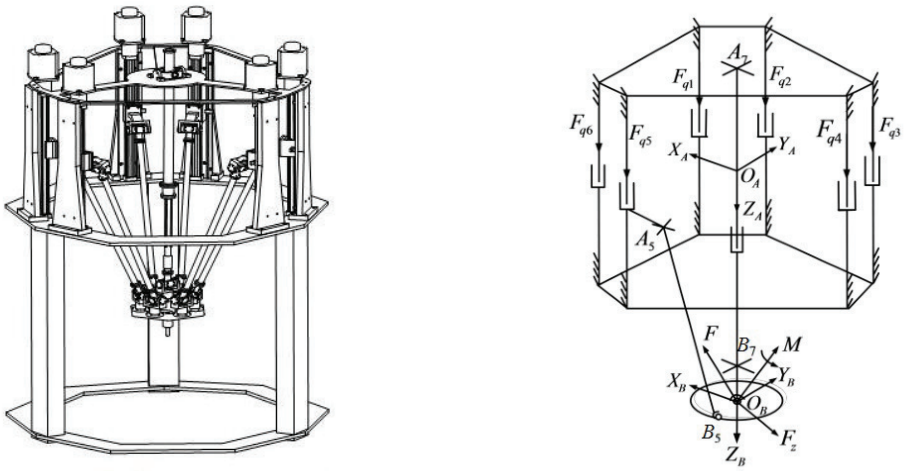
controller has the better robustness property than PI controller. MPC controller can improve the force tracking ability of 6PUS-UPU redundant actuation parallel robot, guarantee the robustness of 6PUS-UPU redundant actuation parallel robot under the parameter variations and load disturbance environment.

This paper is organized as follows. In Section II, we analyze the dynamic model of the 6PUS-UPU redundant actuation parallel robot, and obtain the dynamic mode by KANE method. In Section III, 6PUS-UPU redundant actuation parallel robot force/position hybrid control structure and control design is presented. We build the PMSM predictive control model, and the redundant branch control strategy is designed using PI-PI strategy and PI-MPC strategy respectively. In Section IV, we build the PMSM space voltage vector control simulation platform using the simulink of MATLAB. Then we perform the simulations and present the simulation results. In Section V, a conclusion is drawn.

II. Dynamics modeling

The dynamics modeling of 6PUS-UPU redundant actuation parallel robot will be introduced in this part on the basis of the results of the reference [10], [33]. The modeling process includes velocity analysis, the angular velocity analysis, the constraints analysis and Kane equation.

A. Velocity analysis



(a) 6PUS-UPU Model (b) 6PUS-UPU Coordinate system

Fig. 1: The architecture of 6PUS-UPU redundant actuation parallel robot

6PUS-UPU redundant actuation parallel robot consists of movable platform, fixed platform, six actuators which connect the movable platform and fixed platform, and a constraint branch. Because the middle branch constraints for a degree of freedom of moving platform, moving platform can realize five degrees of freedom of movement, but the robot has six drive, so the robot driver is redundant. The six actuators connects the movable platform and fixed platform through prismatic pair (p), universal joint (u), and spherical hinge (s). The constraint branch connects the two platforms through two universal joints (u). Fig.1 shows the architecture of 6PUS-UPU redundant actuation parallel robot. Fig.1(b) shows the coordinate system of 6PUS-UPU parallel robot. As we can see from Fig.1(b), O_A is the origin of fixed coordinate system, and O_B is the origin of the movable coordinate system. X_A, Y_A, Z_A are the coordinates of fixed coordinate system, X_B, Y_B, Z_B are the coordinates of movable coordinate system. $F_{qi}(i = 1, 2, \cdots 6)$ is the force of six branches. A_5 is the connecting point of connecting rod and fifth branch, B_5 is the connecting point of connecting rod and movable platform. The hinge point between the upper connecting rod of the middle branch and the fixed platform is A_7, the hinge point between the lower connecting rod and the movable platform is B_7. F_z is the constraints of the middle branch for the moving platform. F and M are the external force and torque respectively. Fig.2 shows the universal joint and spherical hinge hinge point distribution diagram.

In this paper, six pose parameters of the moving coordinate system is selected in the fixed coordinates as the generalized coordinates, which are expressed as $\boldsymbol{q} = [x, y, z, \alpha, \beta, \gamma]^T$. $[\alpha, \beta, \gamma]^T$are the Euler angels. $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are the velocity and accelerate vectors of the movable platform respectively. The movable platform velocity vector in the Cartesian coordinate system can be shown as eq.(1).

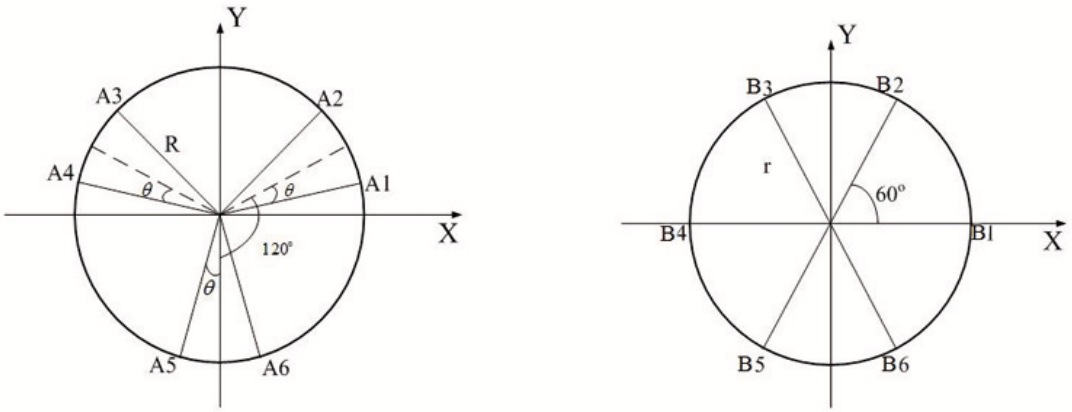
$$(\boldsymbol{v}_d, \boldsymbol{w}_d) = (\dot{x}, \dot{y}, \dot{z}, {}^A w_{Bx}, {}^A w_{By}, {}^A w_{Bz}), \tag{1}$$

The coordinate transforming relationship between the two coordinate systems can be expressed as eq.(2).

$$(\boldsymbol{v}_d, \boldsymbol{w}_d)^T = J_D \dot{\boldsymbol{q}}, \tag{2}$$

where $\boldsymbol{J}_D = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{O}_{3\times3} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{J}_d \end{bmatrix}, \boldsymbol{J}_d = \begin{bmatrix} 1 & 0 & s\beta \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha c\beta \end{bmatrix}$. $\boldsymbol{v}_d$ is the linear velocity, and $\boldsymbol{w}_d$ is the angular velocity.

We can obtain eq.(3) from the Fig.2.



(a) Spherical hinge distribution diagram (b) Universal joint distribution diagram

Fig. 2: Distribution diagram of joint of 6PUS-UPU.

$$\boldsymbol{L}_i = \boldsymbol{B}_i - \boldsymbol{A}_i, (i = 1, 2, \cdots 6) \tag{3}$$

where $\boldsymbol{B}_i$ is the coordinate in the movable coordinate system, $\boldsymbol{B}_i = \boldsymbol{P} + {}^A\boldsymbol{R}_B\boldsymbol{r}_{Bi}, (i = 1, 2, \cdots 6)$, $\boldsymbol{P}$ is the center $\boldsymbol{O}_B$ of the movable platform, ${}^A\boldsymbol{R}_B$ is the transmission matrix of moving coordinate system in fixed coordinate system, $\boldsymbol{r}_{Bi}$ is the location vector of B_i in the movable coordinate system, $\boldsymbol{A}_i$ is the coordinate in the fixed coordinate system.

Without consideration the elastic deformation, the length of the link is a constant L. Then we can obtain eq.(4) [10].

$$L^2 = (B_{ix} - A_{ix})^2 + (B_{iy} - A_{iy})^2 + (B_{iz} - A_{iz})^2, i = 1, 2, \cdots 6 \tag{4}$$

We take the derivative of this equation, then we can get eq.(5).

$$\dot{\boldsymbol{A}}_{iz} = \left(\frac{(\boldsymbol{B}_{ix} - \boldsymbol{A}_{ix})\partial\boldsymbol{B}_{ix}}{(\boldsymbol{B}_{iz} - \boldsymbol{A}_{iz})\partial\boldsymbol{q}} + \frac{(\boldsymbol{B}_{iy} - \boldsymbol{A}_{iy})\partial\boldsymbol{B}_{iy}}{(\boldsymbol{B}_{iz} - \boldsymbol{A}_{iz})\partial\boldsymbol{q}} + \frac{\partial\boldsymbol{B}_{iz}}{\partial\boldsymbol{q}} \right) \dot{\boldsymbol{q}} = \boldsymbol{J}_{Hi}\dot{\boldsymbol{q}} \tag{5}$$

where $i = 1, 2, \cdots 6$.

We can know the velocity of each slider is identical to the velocity A_i based on the features of this machine. We suppose the velocity of each slider along z-axis is $\dot{\boldsymbol{l}}_i$, then $\dot{\boldsymbol{l}}_i = \dot{\boldsymbol{A}}_{iz} = \boldsymbol{J}_{Hi}\dot{\boldsymbol{q}}, i = 1, 2, \cdots 6$.

The velocity of the slider in the fixed coordinate system can be expressed as the following eq.(6).

$$\boldsymbol{v}_{Hi} = \begin{bmatrix} \boldsymbol{O}_{1\times 6} \\ \boldsymbol{O}_{1\times 6} \\ \boldsymbol{J}_{Hi} \end{bmatrix} \cdot \dot{\boldsymbol{q}}, i = 1, 2, \cdots 6 \tag{6}$$

The velocity of B_i is:

$$\boldsymbol{v}_{Bi} = \boldsymbol{v}_d + \boldsymbol{w}_d \times \boldsymbol{r}_{Bi} = \boldsymbol{v}_{Hi} + \boldsymbol{w}_{Li} \times \boldsymbol{n}_i \boldsymbol{L} \tag{7}$$

where $\boldsymbol{r}_{Bi} = {}^A\boldsymbol{R}_B \boldsymbol{r}_{Bi}$ is the position vector of the hinge point B_i in the fixed coordinate system, w_{Li} is the angular velocity of the link, $n_i = (B_i - A_i)/L$ is the unit vector of the link. By the eq.(6) and n_i, we can get the angular velocity of the link is

$$\boldsymbol{w}_{Li} = \frac{\boldsymbol{n}_i \times (\boldsymbol{v}_d + \boldsymbol{w}_d \times \boldsymbol{r}_{Bi} - \boldsymbol{v}_{Hi})}{\boldsymbol{L}} \tag{8}$$

So the linear velocity of the centroid of the link is shown in eq.(9).

$$\boldsymbol{v}_{Li} = \boldsymbol{v}_{Hi} + \boldsymbol{w}_{Li} \times \boldsymbol{n}_i \cdot \frac{\boldsymbol{L}}{2} \tag{9}$$

We write the hinge point between movable platform and the middle constraint (UPU) branch as B_7, then using the movement of movable platform, the velocity of B_7 can be shown as eq.(10).

$$\boldsymbol{v}_{B7} = \boldsymbol{v}_d + \boldsymbol{w}_d \times \boldsymbol{r}_{B7} \tag{10}$$

Using the movement of the middle constraint (UPU) branch, the velocity of B_7 can be shown as eq.(11).

$$\boldsymbol{v}_{B7} = \dot{\boldsymbol{l}}_z \boldsymbol{s} + \boldsymbol{w}_{Lz} \times \boldsymbol{L}_z \tag{11}$$

where $\boldsymbol{r}_{B7} = {}^A\boldsymbol{R}_B \boldsymbol{r}_{B7}$ is the position vector of the hinge point B_7 in the fixed coordinate system, $\boldsymbol{L}_z$ is the vector of the middle branch, $\boldsymbol{L}_z = \boldsymbol{B}_7 - \boldsymbol{A}_7$, $l_z = \sqrt{(B_{7x} - A_{7x})^2 + (B_{7y} - A_{7y})^2 + (B_{7z} - A_{7z})^2}$ is

the length of the middle branch, $\boldsymbol{s}$ is the unit vector of the middle branch, $\boldsymbol{s} = \boldsymbol{L}_z/l_z$; $\boldsymbol{w}_{Lz}$ is the angular velocity of the middle branch, which is shown as eq.(12).

$$\boldsymbol{w}_{Lz} = \frac{\boldsymbol{s} \times (\boldsymbol{v}_d + \boldsymbol{w}_d \times \boldsymbol{r}_{B7})}{l_z} \tag{12}$$

So we can obtain the velocity of the centroid of the upper and lower link of the middle constraint (UPU) branch, which can be expressed as the following eq.(13) and eq.(14).

$$\boldsymbol{v}_{zu} = \boldsymbol{w}_{Lz} \times \boldsymbol{s} l_u \tag{13}$$

$$\boldsymbol{v}_{zl} = \dot{\boldsymbol{l}}_z \boldsymbol{s} + \boldsymbol{w}_{Lz} \times \boldsymbol{s}(l_z - l_l) \tag{14}$$

where l_u is the distance between the hinge point of the fixed platform and the centroid of the upper link, l_l is the distance between the hinge point of the moving platform and the centroid of the lower link, $\dot{\boldsymbol{l}}_z$ is the velocity of relative movement, $\dot{\boldsymbol{l}}_z = \boldsymbol{v}_{B7}\boldsymbol{s}$.

B. Partial velocity and angular velocity analysis

According to the definition of partial velocity and angular velocity, we can get the partial velocity and angular velocity of the movable platform as the following eq.(15) [10].

$$\begin{cases} \boldsymbol{v}_d^* = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O}_3 \end{bmatrix} \\ \boldsymbol{w}_d^* = \begin{bmatrix} \boldsymbol{O}_3 & \boldsymbol{J}_d \end{bmatrix} \end{cases} \tag{15}$$

The partial velocity of the slider is shown in eq.(16).

$$\boldsymbol{v}_{Hi}^* = [\boldsymbol{O}_{1\times6}; \boldsymbol{O}_{1\times6}; \boldsymbol{J}_{Hi}] \tag{16}$$

The partial velocity and angular velocity of the link can be expressed as eq.(17) and eq.(18).

$$\boldsymbol{v}_{Li}^* = \boldsymbol{v}_{Hi}^* + \boldsymbol{w}_{Li}^* \times \boldsymbol{n}_i \cdot \frac{\boldsymbol{L}}{2} \tag{17}$$

$$\boldsymbol{w}_{Li}^* = \frac{\boldsymbol{n}_i \times (\boldsymbol{v}_d^* + \boldsymbol{w}_d^* \times \boldsymbol{r}_{Bi} - \boldsymbol{v}_{Hi}^*)}{\boldsymbol{L}} \tag{18}$$

The partial velocity and angular velocity of the middle constraint (UPU) branch can be expressed as eq.(19) and eq.(20).

$$\begin{cases} \boldsymbol{v}_{zu}^* = \boldsymbol{w}_{Lz}^* \times \boldsymbol{s} l_u \\ \boldsymbol{v}_{zd}^* = \boldsymbol{s}\dot{\boldsymbol{l}}_z^* + \boldsymbol{w}_{iz}^* \times \boldsymbol{s}(l_z - l_d) \end{cases} \tag{19}$$

$$\boldsymbol{w}_{Lz}^{*}=\frac{\boldsymbol{s}\times(\boldsymbol{v}_{d}^{*}+\boldsymbol{w}_{d}^{*}\times\boldsymbol{r}_{B7})}{l_{z}} \tag{20}$$

where $\dot{\boldsymbol{l}}_{z}^{*}=\boldsymbol{s}^{T}\cdot(\boldsymbol{v}_{d}^{*}+\boldsymbol{w}_{d}^{*}\times\boldsymbol{r}_{B7})$ is the partial velocity of $\dot{\boldsymbol{l}}_{z}$.

C. Acceleration analysis

The linear acceleration of the movable platform is shown in the eq.(21).

$$\boldsymbol{a}_{d}=\boldsymbol{v}_{d}^{*}\cdot\ddot{\boldsymbol{q}} \tag{21}$$

The angular acceleration of the movable platform can be expressed as eq.(22).

$$\boldsymbol{\varepsilon}_{d}=\boldsymbol{w}_{d}^{*}\cdot\ddot{\boldsymbol{q}}+\begin{bmatrix}\boldsymbol{O}_{3\times3} & \frac{d\boldsymbol{J}_{d}}{dt}\end{bmatrix}\cdot\dot{\boldsymbol{q}} \tag{22}$$

The linear acceleration of the slider is shown in eq.(23).

$$\boldsymbol{a}_{Hi}=\begin{bmatrix}0 & 0 & \ddot{\boldsymbol{l}}_{Hi}\end{bmatrix}^{T} \tag{23}$$

where $\ddot{\boldsymbol{l}}_{Hi}=\dot{\boldsymbol{J}}_{Hi}\dot{\boldsymbol{q}}+\boldsymbol{J}_{Hi}\ddot{\boldsymbol{q}}$. Using the movement of the moving platform and the movement of each drive branch, the acceleration of B_i can be expressed as eq.(24).

$$\begin{cases}\boldsymbol{a}_{Bi}=\boldsymbol{a}_{d}+\boldsymbol{\varepsilon}_{d}\times\boldsymbol{r}_{Bi}+\boldsymbol{w}_{d}\times(\boldsymbol{w}_{d}\times\boldsymbol{r}_{Bi})\\ \boldsymbol{a}_{Bi}=\boldsymbol{a}_{Hi}+\boldsymbol{\varepsilon}_{Li}\times\boldsymbol{n}_{i}\boldsymbol{L}+\boldsymbol{w}_{Li}\times(\boldsymbol{w}_{Li}\times\boldsymbol{n}_{i}\boldsymbol{L})\end{cases} \tag{24}$$

where $\boldsymbol{\varepsilon}_{Li}$ is the angular acceleration of each drive branch link, $\boldsymbol{\varepsilon}_{Li}=\boldsymbol{n}_{i}\times(\boldsymbol{a}_{Bi}-\boldsymbol{a}_{Hi})/L$. The centroid acceleration of each drive branch link is:

$$\boldsymbol{a}_{Bi}=\boldsymbol{a}_{Hi}+\boldsymbol{\varepsilon}_{Li}\times\boldsymbol{n}_{i}L/2+\boldsymbol{w}_{Li}\times(\boldsymbol{w}_{Li}\times\boldsymbol{n}_{i}L/2). \tag{25}$$

Using the movement of the moving platform and the movement of the middle constraint (UPU) branch to express the acceleration of B_7 are:

$$\begin{cases}\boldsymbol{a}_{B7}=\boldsymbol{\varepsilon}_{Lz}\times\boldsymbol{L}_{z}+\boldsymbol{w}_{Lz}\times(\boldsymbol{w}_{Lz}\times\boldsymbol{L}_{z})+\ddot{\boldsymbol{l}}_{z}\boldsymbol{s}+2\boldsymbol{w}_{Lz}\times\dot{\boldsymbol{l}}_{z}s\\ \boldsymbol{a}_{B7}=\boldsymbol{a}_{d}+\boldsymbol{\varepsilon}_{d}\times\boldsymbol{r}_{B7}+\boldsymbol{w}_{d}(\boldsymbol{w}_{d}\times\boldsymbol{r}_{B7})\end{cases}. \tag{26}$$

The centroid acceleration of the upper and lower link of the middle constraint (UPU) branch is expressed as eq.(27).

$$\begin{cases}\boldsymbol{a}_{zu}=\left[\boldsymbol{\varepsilon}_{Lz}\times\boldsymbol{s}+\boldsymbol{w}_{Lz}\times(\boldsymbol{w}_{Lz}\times\boldsymbol{s})\right]l_{u}\\ \boldsymbol{a}_{zl}=\left[\boldsymbol{\varepsilon}_{Lz}\times\boldsymbol{s}+\boldsymbol{w}_{Lz}\times(\boldsymbol{w}_{Lz}\times\boldsymbol{s})\right](\boldsymbol{l}_{z}-\boldsymbol{l}_{d})+\dot{\boldsymbol{l}}_{z}\boldsymbol{s}+2\boldsymbol{w}_{Lz}\times\dot{\boldsymbol{l}}_{z}\boldsymbol{s}\end{cases} \tag{27}$$

D. Constraints analysis of middle constraint (UPU) branch

We know the middle branch constrain the movable platform from the character of this machine, and the constrain situation is complicated. Based on the screw theory [34], we can get the motion screw system of the middle branch is:

$$\begin{cases} \$_1 = (\ 0 \quad 0 \quad 0\ ;\ 0 \quad 0 \quad 0\ ) \\ \$_2 = (\ \sin\theta_1 \quad -\cos\theta_1 \quad 0\ ;\ 0 \quad 0 \quad 0\ ) \\ \$_3 = (\ 0 \quad 0 \quad 0\ ;\ \cos\theta_1\sin\theta_2 \quad \sin\theta_1\sin\theta_2 \quad -\cos\theta_2\ ) \\ \$_4 = (\ \sin\theta_1 \quad -\cos\theta_1 \quad 0\ ;\ -d_3\cos\theta_1\cos\theta_2 \quad -d_3\sin\theta_1\cos\theta_2 \quad -d_3\cos\theta_2\ ) \\ \$_5 = (\ \cos\theta_1\sin(\theta_2+\theta_4) \quad \sin\theta_1\sin(\theta_2+\theta_4) \quad -\cos(\theta_2+\theta_4)\ ;\ d_3\sin\theta_1\sin\theta_4 \quad -d_3\cos\theta_1 \end{cases}$$

Then the constraints on spiral of the middle branch can be expressed as eq.(28).

$$\$^r = \begin{bmatrix} -\sin(\theta_2+\theta_4)\tan\theta_1/(d_3\sin\theta_4) & \sin(\theta_2+\theta_4)/(d_3\sin\theta_4) & 0\ ;\ 1 & \tan\theta_1 & 0 \end{bmatrix} \tag{28}$$

where θ_1, θ_2, d_3 and θ_4 are joint variables of the middle branch. If $\theta_2+\theta_4 = 0$, the moving platform only move but no rotation, and the constraint of the middle constraint (UPU) branch for the moving platform is torque. If $\theta_2+\theta_4 \neq 0$, the moving platform turns around the x-axis or the y-axis, and the constraint of the middle constraint (UPU) branch for the moving platform is force. In this paper, we regard the constraint as $\boldsymbol{M}_c$ uniformly.

E. KANE Equation

Generalized velocity consists of 6 components. We suppose that each generalized velocity component's generalized force is $\boldsymbol{F}_j^r$, and generalized initial force is $\boldsymbol{F}_j^{*r}$. The driving force of the six branches can be expressed as $\boldsymbol{F}_{qi}$. $\boldsymbol{M}_c$ is the constraints of the middle branch for the moving platform and the external force and torque are $\boldsymbol{F}$ and $\boldsymbol{M}$ respectively. Then we can get eq.(29) and eq.(30) [33].

$$\begin{aligned} \boldsymbol{F}_j^r &= m_d\boldsymbol{g}\boldsymbol{v}_{d,j}^* + \boldsymbol{F}\boldsymbol{v}_{d,j}^* + \boldsymbol{M}\boldsymbol{w}_{d,j}^* + \sum_{i=1}^{6} m_{Hi}\boldsymbol{g}\boldsymbol{v}_{Hi,j}^* + \sum_{i=1}^{6}\boldsymbol{F}_{qi}\boldsymbol{J}_{Hi,j} + \boldsymbol{M}_c\boldsymbol{v}_{d,j}^* \\ &+ \sum_{i=1}^{6} m_{Li}\boldsymbol{g}\boldsymbol{v}_{Li,j}^* + m_{zu}\boldsymbol{g}\boldsymbol{v}_{zu,j}^* + m_{zl}\boldsymbol{g}\boldsymbol{v}_{zd,j}^* \end{aligned} \tag{29}$$

$$\begin{aligned} \boldsymbol{F}_j^{*r} &= -m_d\boldsymbol{a}_d\boldsymbol{v}_{d,j}^* - \sum_{i=1}^{6} m_{Hi}\boldsymbol{a}_{Hi}\boldsymbol{v}_{Hi,j}^* - \sum_{i=1}^{6} m_{Li}\boldsymbol{a}_{Li}\boldsymbol{v}_{Li,j}^* - m_{zu}\boldsymbol{a}_{zu}\boldsymbol{v}_{zu,j}^* \\ &-m_{zl}\boldsymbol{a}_{zl}\boldsymbol{v}_{zl,j}^* - (\boldsymbol{I}_d\boldsymbol{\varepsilon}_d + \boldsymbol{w}_d\times\boldsymbol{I}_d\boldsymbol{w}_d)\boldsymbol{w}_{d,j}^* - (\boldsymbol{I}_{zu}\boldsymbol{\varepsilon}_{Lz} + \boldsymbol{w}_{Lz}\times\boldsymbol{I}_{zu}\boldsymbol{w}_{Lz})\boldsymbol{w}_{Lz}^* \\ &-(\boldsymbol{I}_{zl}\boldsymbol{\varepsilon}_{Lz} + \boldsymbol{w}_{Lz}\times\boldsymbol{I}_{zl}\boldsymbol{w}_{Lz})\boldsymbol{w} - \sum_{i=1}^{6}(\boldsymbol{I}_{Li}\boldsymbol{\varepsilon}_{Li} + \boldsymbol{w}_{Li}\times\boldsymbol{I}_{Li}\boldsymbol{w}_{Li})\,\boldsymbol{w}_{Li,j}^* \end{aligned} \tag{30}$$

where m_d is the mass of the movable platform, m_{Hi} is the mass of each slider, m_{Li} is the mass of each link, m_{zu} is the mass of the upper link of the middle constraint branch, m_{zl} is the mass of the lower link of the middle constraint branch, $\boldsymbol{I}_d$ is the movable platform inertia matrix, $\boldsymbol{I}_{Li}$ is the inertia matrix of each link, $\boldsymbol{I}_{zu}$ and $\boldsymbol{I}_{zl}$ are the inertia matrix of the upper and lower link of the middle constraint branch. Then we can get the KANE equation:

$$\boldsymbol{F}^r + \boldsymbol{F}^{*r} = 0 \tag{31}$$

Then we can derive eq.(32) from eq.(31).

$$\boldsymbol{F}^r + \boldsymbol{F}^{*r} = \boldsymbol{G}\begin{bmatrix} \boldsymbol{F}_{q1} & \boldsymbol{F}_{q2} & \boldsymbol{F}_{q3} & \boldsymbol{F}_{q4} & \boldsymbol{F}_{q5} & \boldsymbol{F}_{q6} & \boldsymbol{M}_c \end{bmatrix}^T - \boldsymbol{F}^T = 0 \tag{32}$$

We separate the part of the driving force from KANE equation, then we can get the general form of the dynamics equation shown in eq.(33).

$$\boldsymbol{G}\boldsymbol{\tau} = \boldsymbol{F}'^T \tag{33}$$

where $\boldsymbol{G}$ IS the Jacobian matrix between the driving forces and platform; $\boldsymbol{\tau}$ is the driving force vector and $\boldsymbol{F}'^T$ is the rest part of the KANE equation.

III. Control Design

The force/position hybrid control structure of 6PUS-UPU redundant actuation parallel robot is designed in this paper, in order to ensure the position accuracy of the parallel robot, optimize the robot internal force at the same time, avoid the stuck phenomenon in the process of running, prevent the large internal force from breaking up parallel mechanism. The former five branches of 6PUS-UPU redundant actuation parallel robot use PMSM three-loop (current loop, speed loop and position loop) control strategy, and the 6^{th} branch (redundant branch) uses PMSM torque control strategy. The traditional PID algorithm is used in PMSM three-loop control structure. Current control uses MPC algorithm instead of traditional PI method and torque control uses PI algorithm in the 6^{th} branch, which can improve the precision of the motor torque control. PI algorithm is the most common method used in motor current control. In general, this method can achieve good control performance, so in this paper we use PI algorithm comparing with the MPC algorithm that we put forward. In this section, we proposed PI and MPC current control algorithm

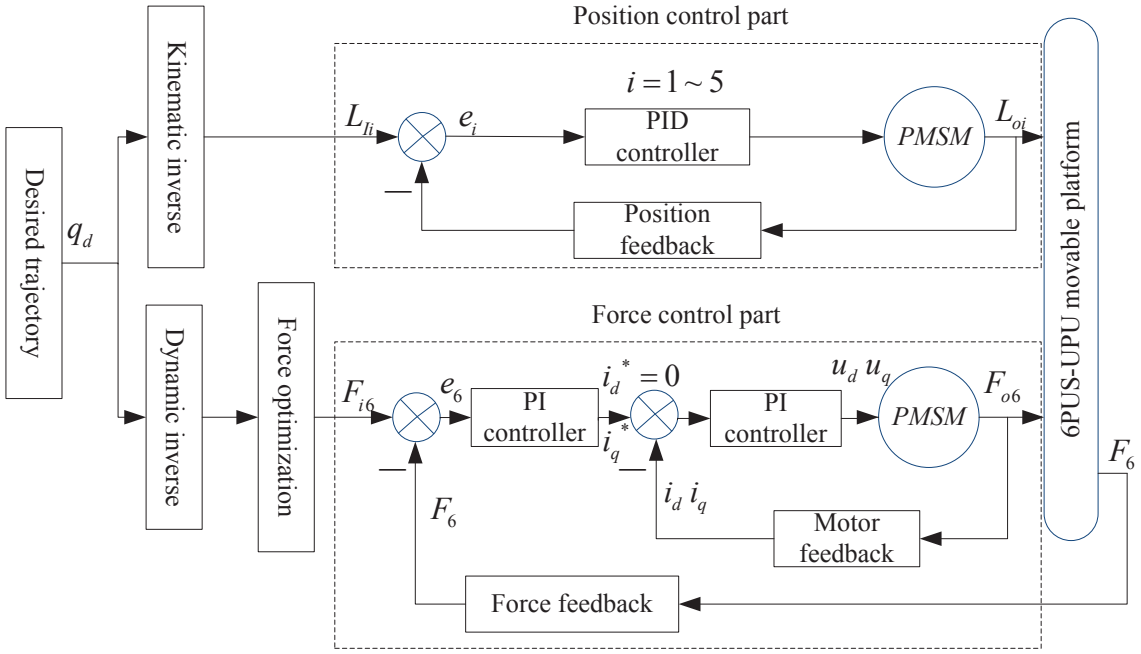
in the 6^{th} branch of the parallel robot. By comparing with the two methods to prove the MPC algorithm has better control performance than PI algorithm in the case of the motor model mismatch.

A. PI Control

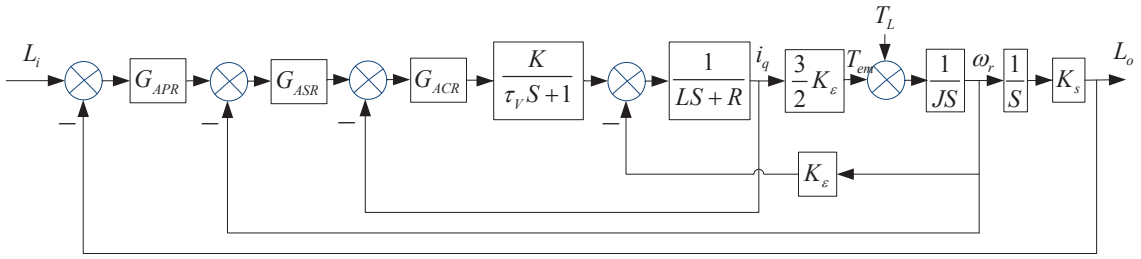
Fig.3(a) is the 6PUS-UPU redundant actuation parallel robot force/position hybrid control strategy structure diagram. Firstly, the position is planned for the actuator on the end of the parallel robot, and then through the kinematics reverse solution to calculate the ideal sliding block position input for the first five position control branches, through the dynamic inverse solution the ideal driving force input for the 6^{th} branch(redundant branch) is calculated. q_d is the position of the actuator. The former five branches is position control part, L_{Ii} is the ideal sliding block position input of the former five branches, $i = 1, 2, 3, 4, 5$. L_{oi} is the actual sliding block position output, $i = 1, 2, 3, 4, 5$. e_i is the sliding block position error. The 6^{th} branch is the force control part, F_{i6} is the ideal driving force input of the 6^{th} branch, F_6 is the actual driving force of the 6^{th} branch connecting rod, e_6 is the force error, ${i_d}^*$ is the direct axis current input, ${i_q}^*$ is the quadrature axis current input, i_d is the direct axis current feedback, i_q is the quadrature axis current feedback. u_d, u_q is the direct axis and quadrature axis voltage, F_{O6} is the motor driving force. F_6 is the driving force of the 6^{th} branch connecting rod to the moving platform along the z axis direction. Both driving force control and current control adopt PI control algorithm.

Fig.3(b) is the control structure diagram of the position branches of the parallel robot, five position control branches is the same. G_{APR}, G_{ASR}, G_{ACR} are the motor position loop, speed loop and current loop controller respectively. The position loop uses P controller, while speed loop and current loop both use PI controller. The $\frac{K}{\tau_V S+1}$ is simplified transfer function of PWM inverter. $\frac{1}{LS+R}$ is simplified transfer function of PMSM. K_s is the ratio coefficient of ball screw, K_ε is the torque coefficient. T_{em} is the PMSM output torque, T_L is the PMSM load torque, ω_r is the angular velocity.

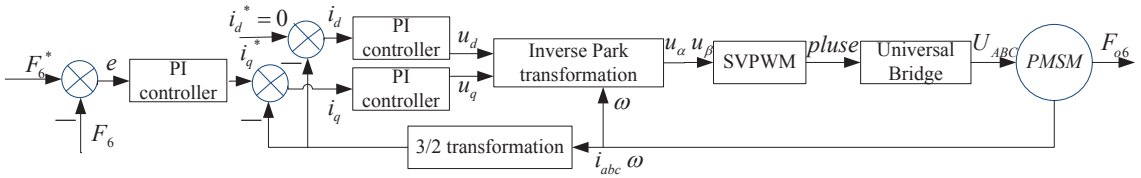
Fig.3(c) is the 6^{th} branch control structure diagram which uses PI current controller. Voltage Space Vector Pulse Width Modulation (SVPWM) technology is used to control the PMSM.



(a) 6PUS-UPU redundant drive parallel robot PI force/position hybrid control diagram



(b) The position control block diagram of the 5 branches of 6PUS-UPU redundant actuation parallel robot

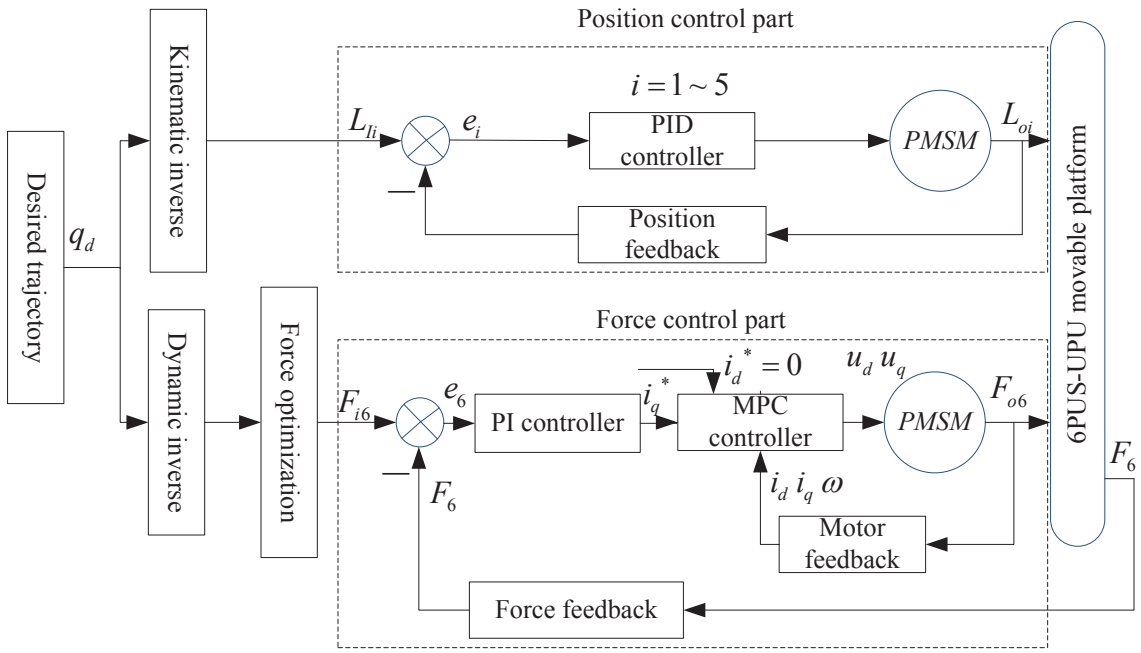


(c) The redundant branch PI control structure block diagram

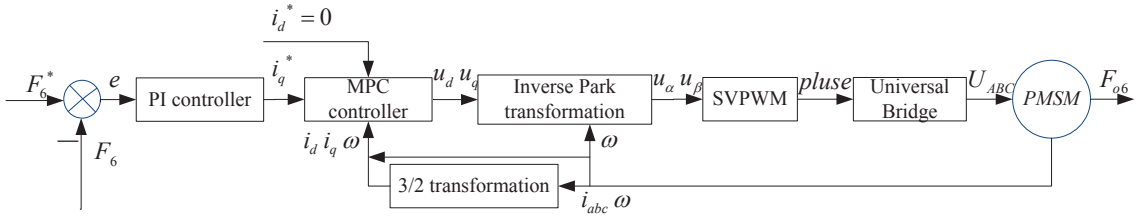
Fig. 3: 6PUS-UPU redundant actuation parallel robot PI control diagram

B. MPC Control

The parameters of the PMSM are easily influenced by factors such as temperature and change in the process of motor running. And the torque control of PMSM requires the motor has a good ability of current tracking, which can ensure the system robustness in the case of the motor parameter change. Therefore, in this paper, we use MPC algorithm instead of the traditional PI regulator to control the current of PMSM,



(a) 6PUS-UPU redundant drive parallel robot MPC force/position hybrid control block diagram



(b) The redundant branch MPC control structure block diagram

Fig. 4: 6PUS-UPU redundant drive parallel robot MPC control strategy

which can improve the tracking accuracy and the robustness of the 6PUS-UPU redundant actuation parallel robot.

Fig.4(a) is the 6PUS-UPU redundant actuation parallel robot force/position hybrid control strategy structure diagram. The former five position control branches are the same as the Fig.3(a), and the 6^{th} branch uses MPC current controller to replace the PI current controller, while the torque control still uses PI controller. Fig.4(b) is the 6^{th} branch control structure diagram, and the current loop adopts MPC controller.

C. MPC controller design

Firstly, we need to establish the the mathematical model of PMSM for the application of MPC. To establish the mathematical model of PMSM, the assumptions are as follows: 1) The reluctance rotor

core is insignificant and can be ignored. The eddy current and hysteresis losses in the premise do not affect significantly the control performance. 2) Permanent magnet conductivity is zero. The inside of the permanent magnet magnetic permeability is the same as the air. 3) No damping windings is considered on the rotor. 4) Permanent magnet excitation magnetic field and three-phase winding armature reaction of the magnetic field in air gap are sine distribution. 5) Phase winding induction electromotive force (EMF) waveform is a sinusoidal wave under steady state operation. 6) Friction coefficient is zero.

In order to obtain the linear state equation, the vector control method $i_d = 0$ is used. On the synchronous dq rotational coordinates, PMSM decoupling state equation is shown as in eq.(34).

$$\begin{bmatrix} \frac{di_d}{dt} \\ \frac{di_q}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & \frac{L_q}{L_d}\omega \\ -\frac{L_d}{L_q}\omega & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix} \begin{bmatrix} u_d \\ u_q \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{\omega\psi_f}{L_q} \end{bmatrix} \tag{34}$$

where L_d and L_q are the quadrature axis inductance and direct axis inductance, respectively, where the unit is H. i_d and i_q are the direct axis current and quadrature axis current, respectively, where the unit is A. u_d and u_q are the direct axis voltage and quadrature axis voltage, respectively, where the unit is V. R_s is the equivalent resistance of motor winding where the unit is Ω. ω is the rotor angular velocity where the unit is rad/s. ψ_f is the equivalent magnetic chain of the rotor magnetic field where the unit is Wb.

Define $A_c = \begin{bmatrix} -\frac{R_s}{L_d} & \frac{L_q}{L_d}\omega \\ -\frac{L_d}{L_q}\omega & -\frac{R_s}{L_q} \end{bmatrix}$, $B_c = \begin{bmatrix} \frac{1}{L_d} & 0 \\ 0 & \frac{1}{L_q} \end{bmatrix}$, $\varepsilon = \begin{bmatrix} 0 \\ -\frac{\omega\psi_f}{L_q} \end{bmatrix}$.

Discretizing model (34), we obtain

$$\begin{cases} x_m(k+1) = A_m x_m(k) + B_m u(k) + \varepsilon \\ y(k) = C_m x_m(k) \end{cases} \tag{35}$$

where $x_m(k) = y(k) = [i_d(k) \;\; i_q(k)]^T$, $u = \begin{bmatrix} u_d(k) & u_q(k) \end{bmatrix}^T$, $A_m = e^{A_c T_s}$, $\mathrm{B}_m = \left(\int_0^{T_s} e^{A_c t} dt\right) B_c$, $C_m = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $T_s = 0.1ms$ is sampling period.

Defining the increment of state $\Delta x_m(k) = x_m(k) - x_m(k-1)$, the increment of control $\Delta u(k) = u(k) - u(k-1)$, then we can get eq.(36) and eq.(37).

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \tag{36}$$

$$y(k+1) = C_m A_m \Delta x_m(k) + y(k) + C_m B_m \Delta u(k) \tag{37}$$

Redefining the system state variables $x(k) = \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$ and using eq.(36) and eq.(37), we can get a new augmented state space equation eq.(38).

$$\begin{cases} \overbrace{\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix}}^{x(k+1)} = \overbrace{\begin{bmatrix} A_m & 0 \\ C_m A_m & I \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}}^{x(k)} + \overbrace{\begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}}^{B} \Delta u(k) \\ y(k) = \overbrace{\begin{bmatrix} 0 & I \end{bmatrix}}^{C} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \end{cases} \tag{38}$$

where $A = \begin{bmatrix} A_m & 0 \\ C_m A_m & I \end{bmatrix}$, $B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}$, $C = \begin{bmatrix} 0 & I \end{bmatrix}$, I is a unit matrix of 2×2, 0 is a zero matrix of 2×2.

According to the working principle of the MPC, the output variables of the system is predicted in the prediction time domain based on the current system status information and state space model. The predictive values of the output variables are deduced as eq.(39).

$$\begin{cases} y(k+1|k) = CAx(k|k) + CB\Delta u(k|k) \\ y(k+2|k) = CA^2 x(k|k) + CAB\Delta u(k|k) + CB\Delta u(k+1|k) \\ \quad \vdots \\ y(k+N_p|k) = CA^{N_p} x(k|k) + CA^{N_p-1} B\Delta u(k|k) + \cdots + \\ \qquad\qquad CA^{N_p-N_c} B\Delta u(k+N_c-1|k) \end{cases} \tag{39}$$

where N_c is the control time domain, and $N_c \leq N_p$, $y(k+j|k)$ is the output variables predictive value of the $k+j$ moment, $j = 1, \cdots, N_p$, $u(k+j|k)$ is the control variables predictive value of the $k+j$ moment, $j = 0, 1, \cdots, N_c - 1$, in the case of $N_c \leq j \leq N_p$, $u(k+j|k) = u(k+N_c-1|k)$.

Thus it can be seen that all the output predictive values are based on the current moment state variables of $x(k)$ and the future control variables $\Delta u(k+j|k)$.

Defining the vector $Y_m = \begin{bmatrix} y(k+1|k)^{\mathrm{T}} & y(k+2|k)^{\mathrm{T}} & y(k+3|k)^{\mathrm{T}} & \cdots & y(k+N_p|k)^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, so eq.(39) can be written as vector form.

$$Y_m = Fx(k) + \Phi\Delta U \tag{40}$$

where $\Delta U = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix}$, $F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}$,

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & & & & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \cdots & CA^{N_p-N_c}B \end{bmatrix}.$$

In order to introduce feedback correction, the closed-loop prediction output equation is shown as in eq.(41).

$$Y_p(k+i) = Y_m(k+i) + He\,(k) \tag{41}$$

where $e\,(k) = y(k) - y_m(k)$ is the output error. H is the error coefficient.

D. Optimal Control

In order to realize the optimal control, the quadratic objective function is used.

$$J = (W - Y_p)^T Q (W - Y_p) + \Delta U^T R \Delta U \tag{42}$$

where Q is the controlled variable weight matrix, its dimension is $(2 \times N_p) \times (2 \times N_p)$, R is the control variable weight matrix, its dimension is $(2 \times N_c) \times (2 \times N_c)$. α_{id}, α_{iq}, α_{ud} and α_{uq} are the weight coefficients of controlled variables i_d and i_q, and the control variables u_d and u_q, respectively. So Q and R matrices can be expressed as $Q = \begin{Bmatrix} \alpha_{id} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & \alpha_{iq} & 0 & \ddots & \cdots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \alpha_{id} & 0 \\ 0 & \cdots & \cdots & 0 & 0 & \alpha_{iq} \end{Bmatrix}$, $R = \begin{Bmatrix} \alpha_{ud} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & \alpha_{uq} & 0 & \ddots & \cdots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \alpha_{ud} & 0 \\ 0 & \cdots & \cdots & 0 & 0 & \alpha_{uq} \end{Bmatrix}$.

Substituting eq.(41) into eq.(42), we can obtain the closed loop form of performance index function eq.(43).

$$\begin{aligned} J = &(W - Fx(k) - He)^T Q\,(W - Fx(k) - He) - 2\Delta U^T \Phi^T Q\,(W - Fx(k) - He) \\ &+\Delta U^T(\Phi^T Q \Phi + R)\Delta U \end{aligned} \tag{43}$$

$$\frac{\partial J}{\partial \Delta U} = 0$$

Then the optional control is

$$\Delta U = (\Phi^T Q \Phi + R)^{-1}\Phi^T Q(W - Fx(k) - He\,(k)). \tag{44}$$

According to the rolling optimization strategy, the first element $\Delta u(k)$ is calculated. The control variable $u(k)$ is the actual control effort on the system at the k moment, which is obtained by eq.(45). At the $k+1$ moment, the latest information $x(k+1)$ of the system is measured. The above optimization process is repeated.

$$u(k) = u(k-1) + \Delta u(k) \tag{45}$$

The above introduces the basic principle of MPC algorithm. Then we describe the online computation process of MPC algorithm. The online computation process of MPC algorithm is made up of initialization module and real-time control module. Initialization module is to set the initial value, and detect the actual output $y(k)$ of the object in the first step of the system is running. And set it as the predict initial value $y_0(k+i|k)$, $i = 1, \ldots, N$. Since the second step in the real-time control module, the online computation process at each sampling time mainly includes the following steps.

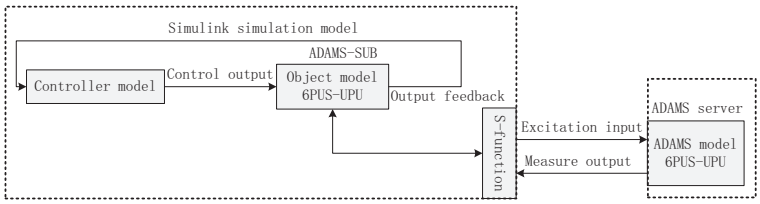
1) Detect the actual output and calculation error : $y - y(1) \to e$.

2) Predictive value correction : $y(i) + h_i e \to y(i), i = 1, \cdots N$.

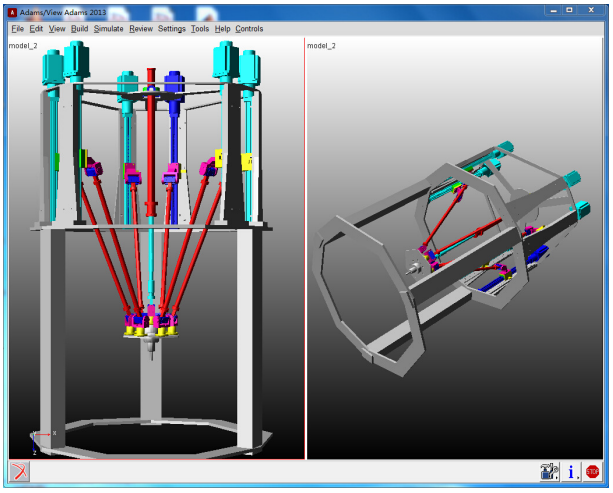
3) Compute the control increment : $\Delta U = (\Phi^T Q \Phi + R)^{-1}\Phi^T Q(W - Fx(k) - He\,(k)) \to \Delta u$.

4) Compute the control input : $u(i-1) + \Delta u \to u$.

5) Compute the output predictive value : $CAx(i-1) + CB\Delta u(i-1) \to y(i), i = 1, \cdots, N$.



(a) MATLAB/ADAMS simulation diagram



(b) The virtual model of 6PUS-UPU redundant actuation parallel robot

Fig. 5: 6PUS-UPU redundant actuation parallel robot simulation platform

IV. Simulation

In this section, we use MATLAB/ADAMS joint simulation to test the validity of the proposed method. The output of MATLAB is defined as the input of ADAMS, while the output of ADAMS is the input of MATLAB. Fig.5(a) is the whole diagram of MATLAB/ADAMS joint simulation. Fig.5(b) is the virtual model of the 6PUS-UPU redundant actuation parallel robot in ADAMS. The motor parameters are shown in Table I. The 6PUS-UPU redundant actuation parallel robot model parameters are shown in Table II.

Firstly, the PMSM space voltage vector control simulation platform is built in the simulink of MATLAB, PI control strategy and MPC strategy are used to validate current and torque tracking performance. Simulation platform are shown in Fig.6(a) and Fig.6(b).

Considering the size of the data of the torque control of the parallel robot, design experiments by adding 30N input torque on the system at 0.01s. The dramatic change in the process of current and torque will be a huge challenge for the current controller. In this paper, we use the method to combine adopting

TABLE I: The parameters of PMSM

Parameter	Meaning	value	Unit
K_S	proportionality coefficient of screw ball	$5/\pi$	/
L	rotator inductance	0.0027	H
R	rotator resistance	1.3	Ω
P_n	number of pairs	1	/
J	movement of initial	0.000188	$Kg \cdot m^2$
K_e	torque constant	0.167	/
$\boldsymbol{\tau}_v$	inverter time constant	0.0001	s
K_n	inverter gain	4.43	/

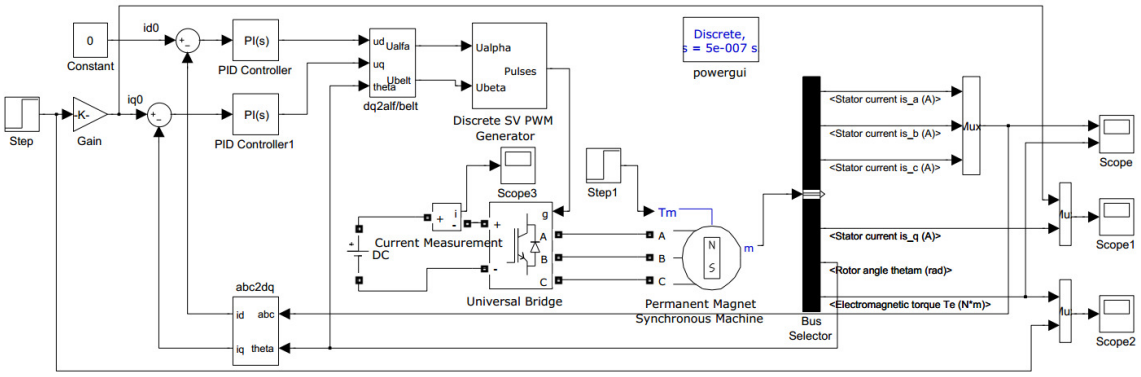
TABLE II: The parameters of 6PUS-UPU redundant actuation parallel robot

Parameter	Meaning	value	Unit
m_d	movable platform mass	32.65	kg
m_H	slider mass	6.4	kg
m_L	link mass	5.8	kg
L	length of link	371	mm
R	radius of fixed platform	500	mm
r	radius of movable platform	135	mm
m_{zu}	mass of the upper link of middle branch	4.18	kg
m_{zd}	mass of the lower link of middle branch	6.15	kg
l_u	length of the upper link of middle branch	311	mm
l_d	length of the lower link of middle branch	448	mm

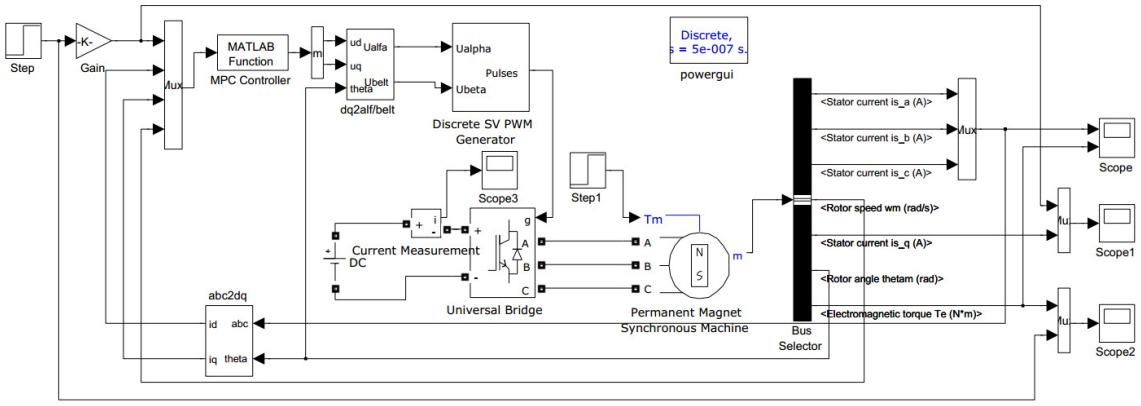
experience value with many times adjustment makes the control performance best to select the controller parameters of MPC controller and PI controller.

In the case of motor parameters matching, the parameters of PI controller are adjusted until the motor obtains the optimal torque response curve, which is P = 0.1 and I = 2.5. The parameters of MPC controller are N_p = 5, N_c = 2, $\alpha_{id} = 9$, $\alpha_{iq} = 9$, $\alpha_{ud} = 2$ and $\alpha_{uq} = 2$. $T_s = 0.1ms$ the is sampling period. Fig.6(a) and Fig.6(b) are the current and torque tracking curve of the PMSM using the PI current controller and MPC controller respectively in the case of model matching.

Fig.7 shows that both PI current controller and MPC current controller can accurately track the input of current and torque. However the current and torque response curves have obvious overshoot when PI controller is used. At 0.02s moment, the system can generally track the desired value. There is a small



(a) PI controller of permanent magnet synchronous motor



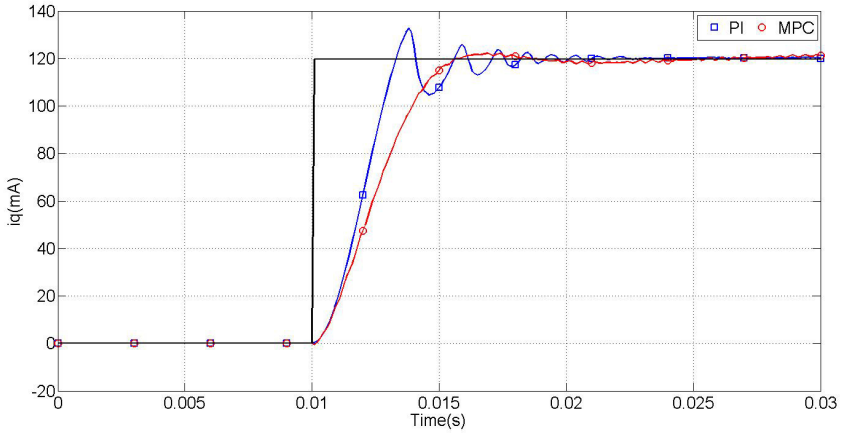
(b) MPC controller of permanent magnet synchronous motor

Fig. 6: Permanent magnet synchronous motor with PI controller and MPC controller in MATLAB/Simulink simulation

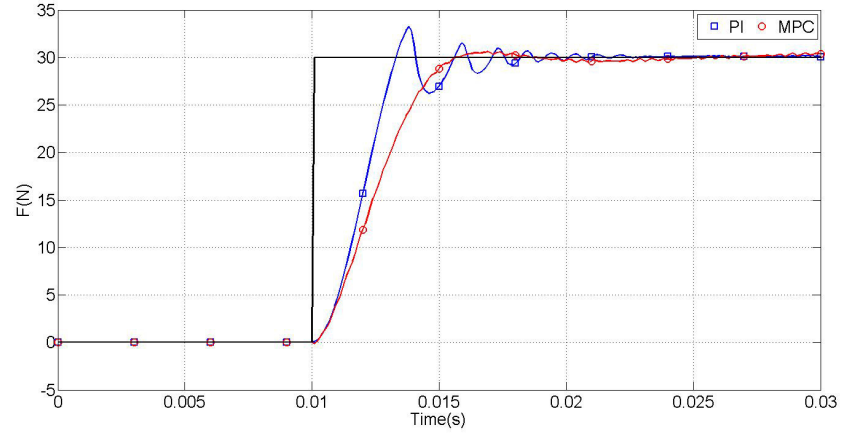
overshoot for MPC controller. At 0.015s moment the system basically can track the desired value, and the tracking speed is much faster than that of PI controller.

In the case of model mismatch, we compare the robustness of PI controller and MPC controller. The motor inductance is 0.9 times as large as the nominal value. Winding resistance is 1.2 times larger than the nominal value. Rotor flux is 1.1 times the size of the nominal value. All the control parameters are consistent with the model parameters matching.

Fig.8 shows the current and torque response curves in the case of model mismatch. It shows that the tracking performance of the current and torque in the case of model mismatch is consistent with model matching by using MPC controller. Both current and torque response curves are without overshoot that they are able to track the desired value quickly. However there is obvious steady-state error of the current



(a) Current tracking curve of PI control algorithm and MPC control algorithm



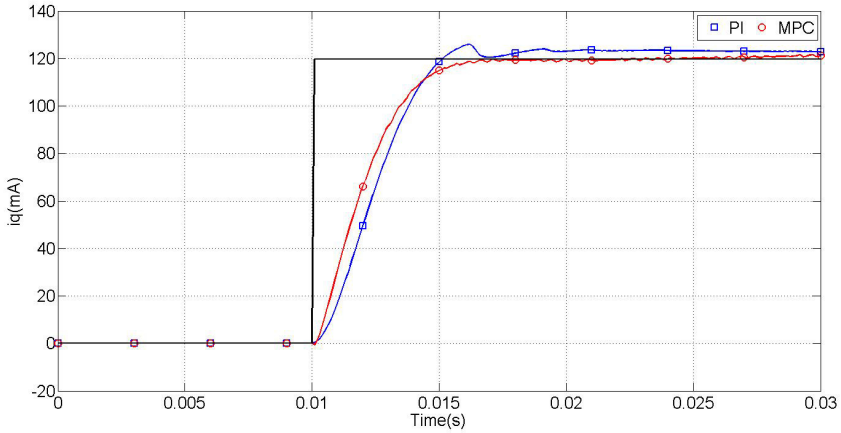
(b) Torque tracking curve of PI control algorithm and MPC control algorithm

Fig. 7: Current and torque tracking curve under model matching

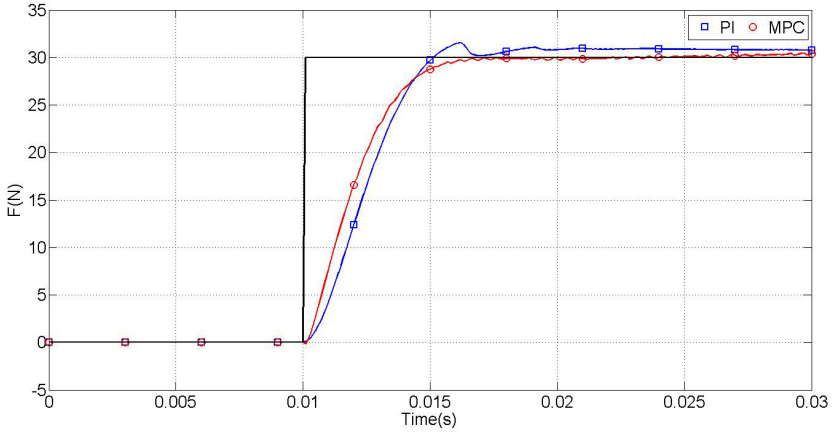
and torque response curves when PI controller is used, and the tracking capability of the current and torque is worse than that of MPC controller.

PI controller and MPC controller are used in MATLAB/ADAMS joint simulation of the 6PUS-UPU redundant actuation parallel robot. The desired trajectory of the platform is designed to move from (0, 0, 928.5273) to $(-100, -100, 928.5273)$, and then move from $(-100,-100, 928.5273)$ to (100, 100, 928.5273), where the unit is mm. According to the desired trajectory of moving platform, the ideal slider positions input of the 1^{st} to 5^{th} branches are calculated by using the kinematics inverse solution and the ideal driving force input of the redundant branch is calculated by using the dynamic inverse solution [10].

Fig.9 shows the ideal driving force input curve of the six branches of the 6PUS-UPU redundant actuation



(a) Current tracking curve of PI control algorithm and MPC control algorithm



(b) Torque tracking curve of PI control algorithm and MPC control algorithm

Fig. 8: Current and torque tracking curve under model mismatch

parallel robot. Fig.10(a) and Fig.10(b) are the actual driving force curves of the six branches by using the PI controller and MPC controller of the 6PUS-UPU redundant actuation parallel robot under model mismatch. Fig.11(a) and Fig.11(b) show the driving force error of the six branches of the 6PUS-UPU redundant actuation parallel robot by using PI controller and MPC controller under model mismatch. Fig.12 and Fig.13 show the slider position curves and the slider position error curves of the five position branches of the 6PUS-UPU redundant actuation parallel robot by using MPC controller under model mismatch.

To compare the performance of MPC controller with PI controller intuitively, we define the average amplitude error evaluation criteria to analyze the six branches torque error of the parallel robot.

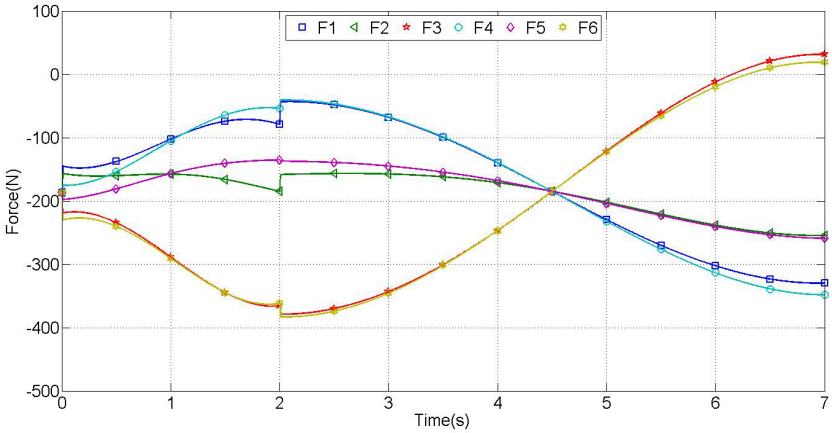
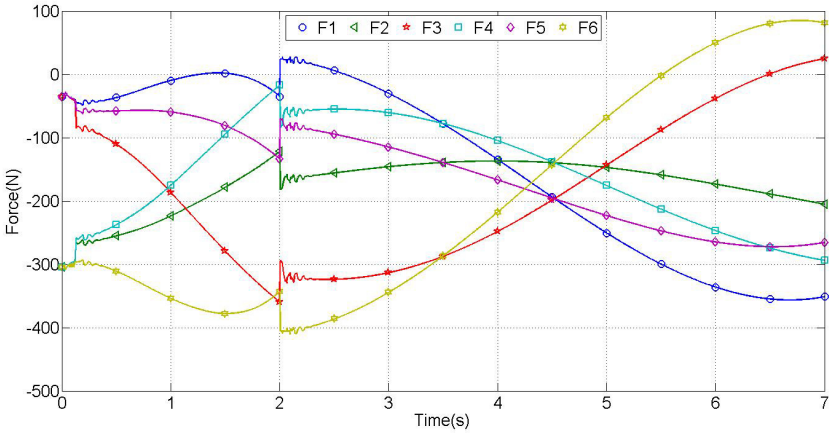
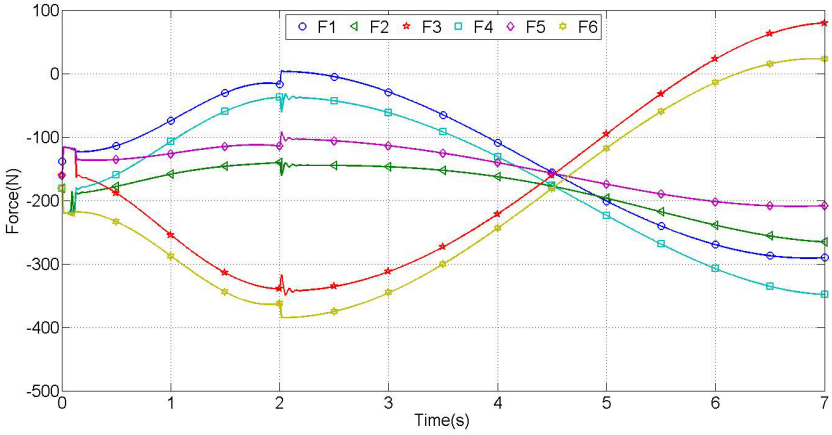


Fig. 9: The ideal driving force of six branches of 6PUS-UPU redundant actuation parallel robot

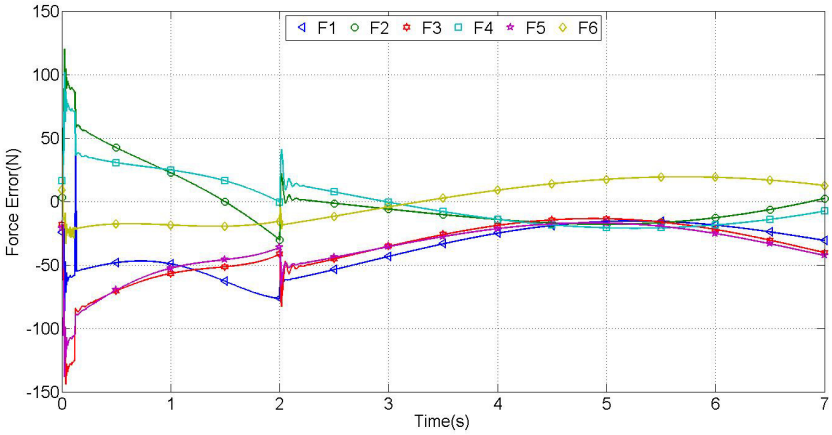


(a) The driving force of parallel robot's six branches with PI controller under model mismatch

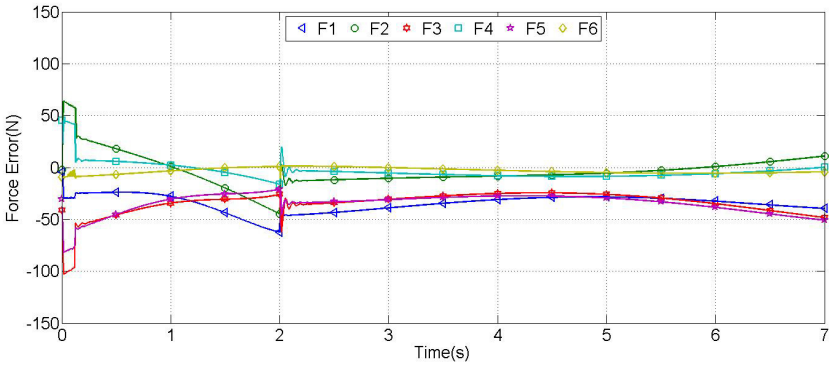


(b) The driving force of parallel robot's six branches with MPC controller under model mismatch

Fig. 10: The driving force of 6PUS-UPU redundant actuation parallel robot's six branches under model mismatch



(a) The driving force error of parallel robot's six branches with PI controller under model mismatch



(b) The driving force error of parallel robot's six branches with MPC controller under model mismatch

Fig. 11: The driving force error of 6PUS-UPU redundant actuation parallel robot's six branches under model mismatch

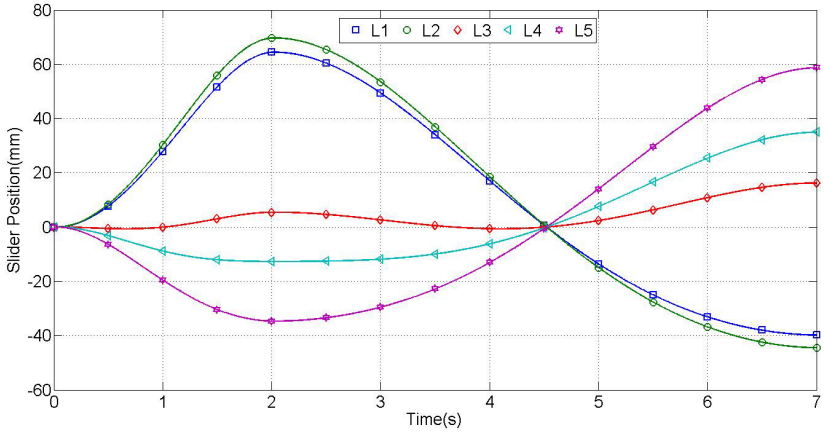


Fig. 12: The slider position of the five position branches of 6PUS-UPU redundant actuation parallel robot with MPC controller under model mismatch

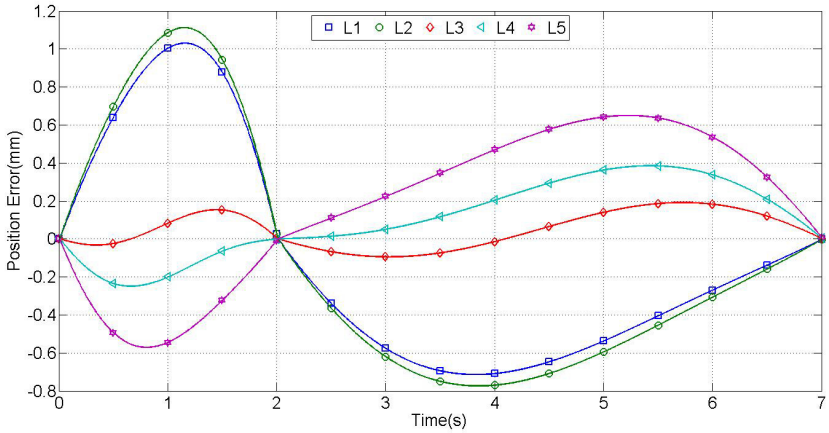


Fig. 13: The slider position error of the five position branches of 6PUS-UPU redundant actuation parallel robot with MPC controller under model mismatch

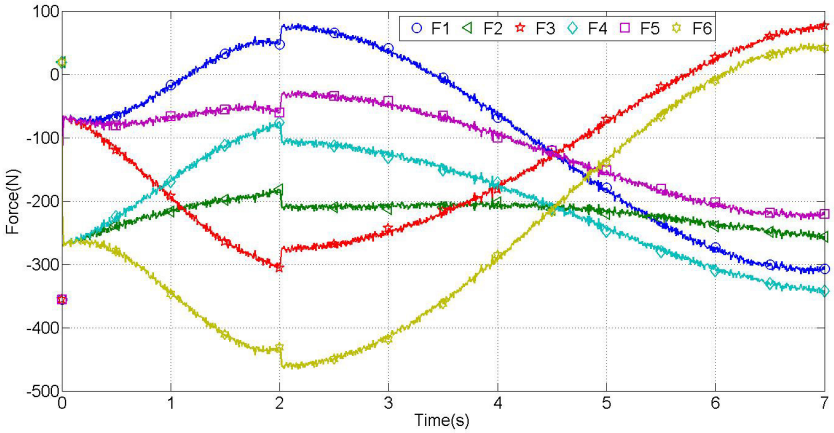
$$I_{pp} = \frac{|E_{MPC} - E_{PI}|}{|E_{PI}|} \tag{46}$$

where E_{MPC} is the torque error average value of the parallel robot using MPC controller, E_{PI} is the torque error average value of the parallel robot using PI controller, I_{PP} describes the improvement of the controllers. The driving force control precision contrast of MPC controller and PI controller is shown in Table III.

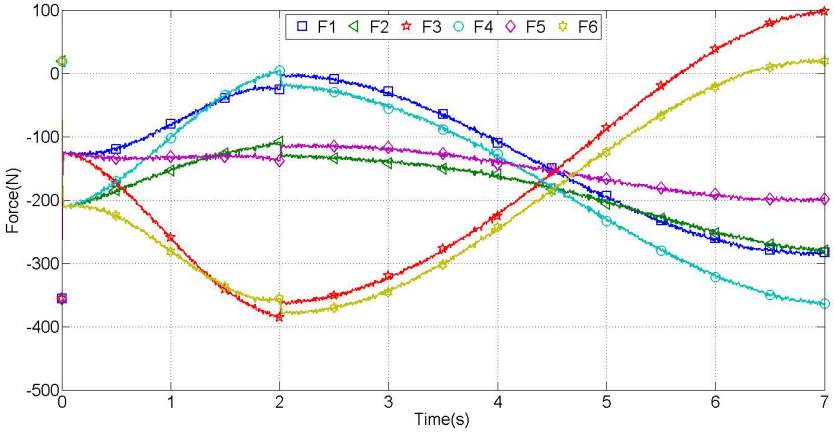
TABLE III: The improvement of the MPC controller than the PI controller

Branch	E_{MPC}(N)	E_{PI}(N)	I_{PP}(%)
1	33.96	37.80	10.16
2	11.85	18.37	35.49
3	32.81	40.86	19.70
4	6.18	19.95	69.02
5	33.34	38.91	14.32
6	4.13	14.87	72.23

Table III shows that the performance of the 6PUS-UPU redundant actuation parallel robot with MPC algorithm is obviously better than that of PI algorithm. The driving force error of six branches of the 6PUS-UPU redundant actuation parallel robot is much smaller than that of PI controller. Fig.13 shows that the slider position error of the five position branches is within 1mm when MPC algorithm is used.



(a) The driving force of parallel robot's six branches with PI controller under disturbance

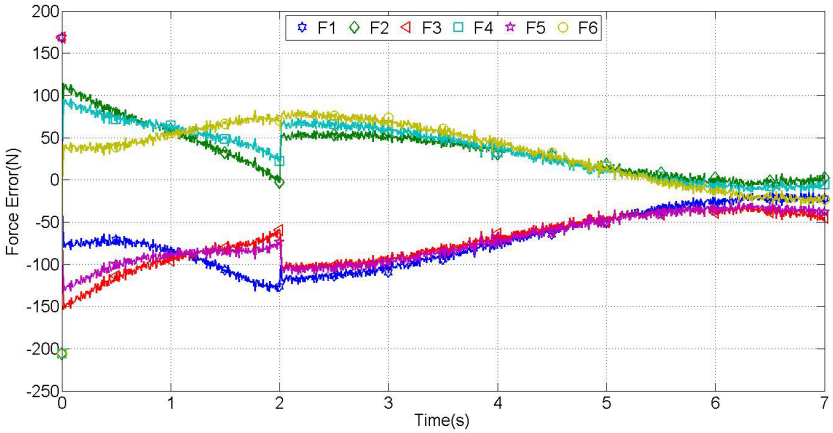


(b) The driving force of parallel robot's six branches with MPC controller under disturbance

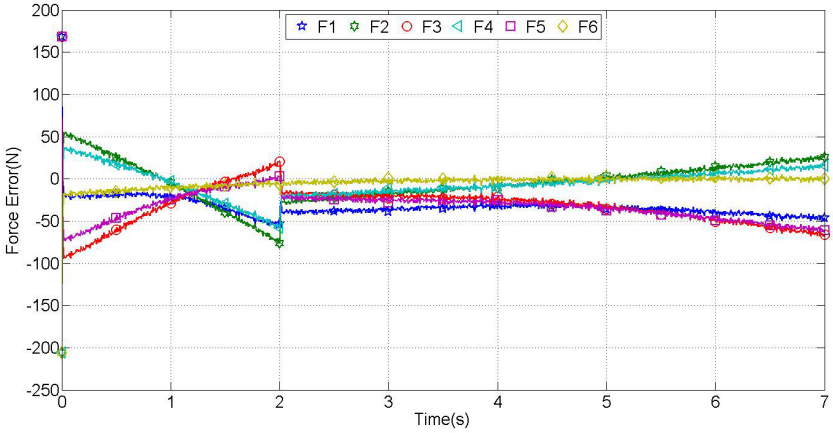
Fig. 14: The driving force of 6PUS-UPU redundant actuation parallel robot's six branches under disturbance

So the 6PUS-UPU redundant actuation parallel robot can obtain higher precision of movement by using MPC algorithm.

We have tested the robustness of the control by adding external disturbances. Fig.14(a) and Fig.14(b) show the driving force curves of the six branches by using the PI controller and MPC controller of the 6PUS-UPU redundant actuation parallel robot under disturbance. Fig.15(a) and Fig.15(b) show the driving force error of the six branches of the 6PUS-UPU redundant actuation parallel robot by using PI controller and MPC controller under disturbance. We can conclude from Fig.14 and Fig.15 that the force control curve of MPC controller is relatively smoother than that of PI controller under disturbance, and the force



(a) The driving force error of parallel robot's six branches with PI controller under disturbance



(b) The driving force error of parallel robot's six branches with MPC controller under disturbance

Fig. 15: The driving force error of 6PUS-UPU redundant actuation parallel robot's six branches under disturbance

error of all branches of 6PUS-UPU redundant actuation parallel robot became small, while the force control precision of all branches with PI controller is significantly lower than that of all branches with MPC controller.

We can conclude that the MPC algorithm in the current loop of PMSM can achieve better torque tracking performance than PI controller. MPC algorithm demonstrates a better robustness property. Furthermore, MPC algorithm used in the redundant branch of the 6PUS-UPU redundant actuation parallel robot can realize the purpose of optimizing system structure internal force and improve the control precision.

V. Conclusion

In this paper, we proposed an MPC algorithm in the PMSM current loop of the parallel robot redundant branch, other five branches use the slider position PID control method. The proposed method not only ensures the dynamic tracking performance of the robot, but also optimizes the structure internal force of the parallel robot. The proposed method prevents the agency internal force from too large which leads to permanent damage to the parallel robot. The system also can obtain good dynamic tracking performance by using MPC algorithm no matter does the case of model match or not. Simulation results show that the proposed method compared with the traditional PI method can improve the driving force tracking performance of the parallel robot and has better robustness. But the MPC algorithm has a more complicated calculation process and its simulation is relatively slow. As a result, our future work is to improve the MPC algorithm, trying to simplify the calculation process.

ACKNOWLEDGMENTS

The work was partly supported by the National Natural Science Foundation of China (Project No.51275439, No.61473248), and the Major State Basic Research Development Program of China 973 program (Project No.2013CB733000), the Natural Science Foundation of Hebei Province of China under the project No.F2014203095, China Postdoctoral Science Foundation(Project No. 2014M560196), Scholars Studying Abroad Science and Technology Activities of Hebei Province of China(Project No.C201400355), the Young Teacher of Yanshan University under the project No.13LGA007.

References

[1] E. Conkur and R. Buckingham, "Clarifying the definition of redundancy as used in robotics," *Robotica*, vol. 5, no. 15, pp. 583–586.

[2] J. GallardoAlvarado, C. AguilarNjera, and L. CasiqueRosas, "Kinematics and dynamics of 2 (3-RPS) manipulators by means of screw theory and the principle of virtual work," *Mechanism and Machine Theory*, vol. 10, no. 43, pp. 1281–1294, 2008.

[3] J. Wang, Gosselin, and C. M, "Kinematic analysis and design of kinematically redundant parallel mechanisms," *ASME J Mech Des*, vol. 1, no. 126, pp. 109–118, 2004.

[4] S. Kock and W. Schumacher, "Redundant parallel kinematic structures and their control," *Springer Tracts in Advanced Robotics*, vol. 6, no. 67, pp. 143–157, 2011.

[5] B. Yi, R. Freeman, and D. Tesar, "Open-loop stiffness control of over-constrained mechanisms/robot linkage systems," in *Proc IEEE Int Conf Robotics Automation*, (Scottsdale AZ).

[6] S. Nokleby, R. Fisher, R. Podhorodeski, and F. Firmani, "Force capabilities of redundantly actuated parallel manipulators," *Mech Mach Theory*, vol. 5, no. 40, pp. 578–599, 2005.

[7] S. Kim, "Operational quality analysis of parallel manipulators with actuation redundancy," in *Proc of IEEE int conf on robotics and automation*, pp. 2651–2656, 1997.

[8] D. Chakarov, "Study of the antagonistic stiffness of parallel manipulators with actuation redundancy," *Mech Mach Theory*, vol. 6, no. 39, pp. 583–601, 2004.

[9] G. M. Mohamed, M. Clment, and Gosselin, "Design and analysis of kinematically redundant parallel manipulators with configurable platforms," *IEEE Trans Robot*, no. 21, pp. 277–287, 2005.

[10] L. Shunpan, D. Yanbin, and Z. Kuijing, "Kinematics control of a new 6PUS-UPS/UPU parallel manipulator with variable mobility," *Proceedings of 2010 International Conference on Mechanic Automation and Control Engineering .Washington,D.C., USA : IEEE*, pp. 6330–6334, 2010.

[11] Q. Xu, "Precision position/force interaction control of a piezoelectric multimorph microgripper for microassembly," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 503–514, 2013.

[12] JK Mills, AA Goldenberg, "Force and position control of manipulators during constrained motion tasks," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 30–46, 1989.

[13] Z. Chen, J. Geng, and X. Liu, "An integral and exponential time-varying sliding mode control of permanent magnet synchronous motors," *Transactions of China Electrotechnical Society*, vol. 6, no. 26, pp. 56–61, 2011.

[14] J. Huang, B. Zhou, and D. Li, "Sliding mode control for permanent magnet synchronous motor servo system," *Transactions of China Electrotechnical Society*, vol. 11, no. 24, pp. 41–47, 2009.

[15] H. Wang, B. Zhou, and S. Fang, "A PMSM sliding mode control system based on exponential reaching law," *Transactions of China Electrotechnical Society*, vol. 9, no. 24, pp. 71–77, 2009.

[16] R. Chen, Z. Deng, and Y. Yan, "Application research on differential feedback control in permanent magnet motor servo system," *Transactions of China Electrotechnical Society*, vol. 9, no. 20, pp. 92–97, 2005.

[17] G. Li, H. Ge, and T. Liu, "Pseudo derivative feedback control for PMSM drive system," *Transactions of China Electrotechnical Society*, vol. 8, no. 25, pp. 18–23, 2010.

[18] H. Zhu, X. Xiao, and Y. Li, "PI type dynamic decoupling control scheme for PMSM high speed operation," in *Proceedings of IEEE Applied Power Electronics Conference and Exposition(APEC)*, pp. 1736–1739, 2010.

[19] Y. Li and M. Zhang, "Status of high performance ac permanent magnet synchronous motor servo system," *Servo Control*, vol. 1, pp. 34–37, 2008.

[20] F. Morel, X. L. Shi, J. M. Rtif, and B. Allard, "A predictive current control applied to a permanent magnet synchronous machine, comparison with a classical direct torque control," *Electric Power Systems Research*, no. 78, pp. 1437–1447, 2008.

[21] R. Jan, C. Tseng, and R. Liu, "Robust PID control design for permanent magnet synchronous motor: A genetic approach," *Electric Power Systems Research*, no. 78, pp. 1161–1168, 2008.

[22] B. Ding, *Modern predictive control*. CRC Press, New York, 2010.

[23] F. A. Lara-Molina, J. M. Rosrio, D. Dumur, and P. Wenger, "Application of predictive control techniques within parallel robot," *Revista Controle and Automacao*, vol. 23, no. 5, pp. 530–540, 2012.

[24] P. Zheng, Y. Xi, and D. Li, "An improved robust model predictive control approach to systems with linear fractional transformation perturbations," *Automation and Computing*, vol. 8, no. 1, pp. 134–140, 2011.

[25] S. K. Kim, D. K. Choi, K. B. Lee, and Y. I. Lee, "Offset-free model predictive control for the power control of three-phase AC/DC converters," *Robotics and Autonomous Systems*, pp. 1–12, 2015.

[26] X. Lin, L. Xie, and H. Su, "Economic performance for predictive control systems under model uncertainty," *Acta Automatica Sinica*, vol. 39, pp. 1141–1145, 2013.

[27] Z. Li and J. Sun, "Disturbance compensating model predictive control with application to ship heading control," *IEEE Transactions on Control Systems Technology*, vol. 20, pp. 257–265, 2012.

[28] V. L. Bageshwar, W. L. Garrard, and R. Rajamani, "Model predictive control of transitional maneuvers for adaptive cruise control vehicles," *IEEE Transactions on Vehicular Systems Technology*, vol. 53, no. 5, pp. 1573–1584, 2004.

[29] J. Scoltock, T. Geyer, and U. K.Madawala, "A comparison of model predictive control schemes for MV induction motor drives," *Robotics and Autonomous Systems*, vol. 9, pp. 909–919, 5 2013.

[30] M. Preindl and S. Bolognani, "Model predictive direct speed control with finite control set of PMSM drive systems," *IEEE Transactions on Vehicular Systems Technology*, vol. 28, no. 2, pp. 1007–1015, 2013.

[31] A.A.Hassan and A. M.Kassem, "Modeling, simulation and performance improvements of a PMSM based on functional model predictive control," *Electrical Engineering*, vol. 38, no. 11, pp. 3071–3079, 2013.

[32] S. C. Carpiuc and C. Lazar, "Energy-efficient model predictive speed control of permanent magnet synchronous machine based automotive traction drives," *Robotics and Autonomous Systems*, pp. 1–6, 2014.

[33] S. H. Gao, "Dynamics modeling and control research of 6pus-upu parallel manipulator," *Yan shan University, China*, 2013.

[34] M. B. A. Khoukhi, L. Baron, "Constrained multi-objective trajectory planning of parallel kinematic machines," *Robotics and Computer-Integrated Manufacturing*, vol. 25, pp. 756–769, Aug. 2009.